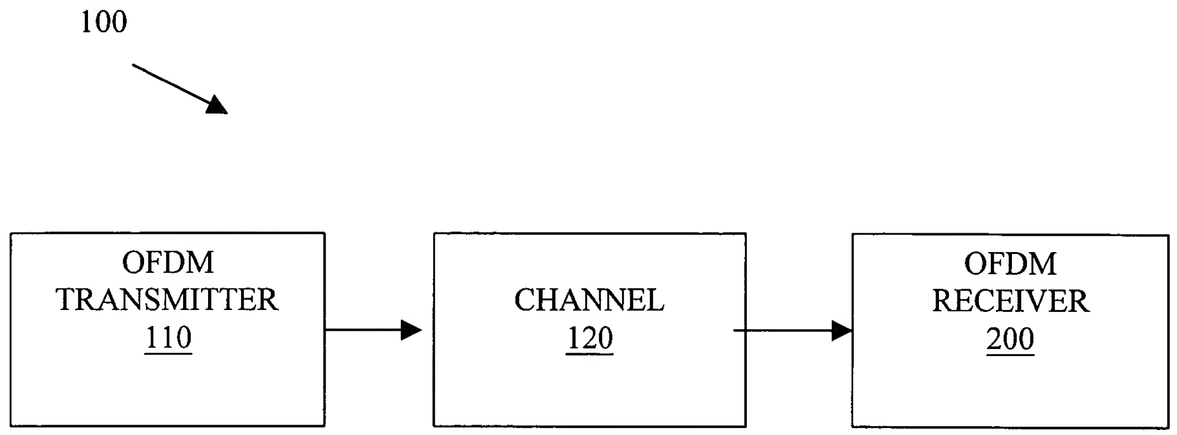


FIG. 1

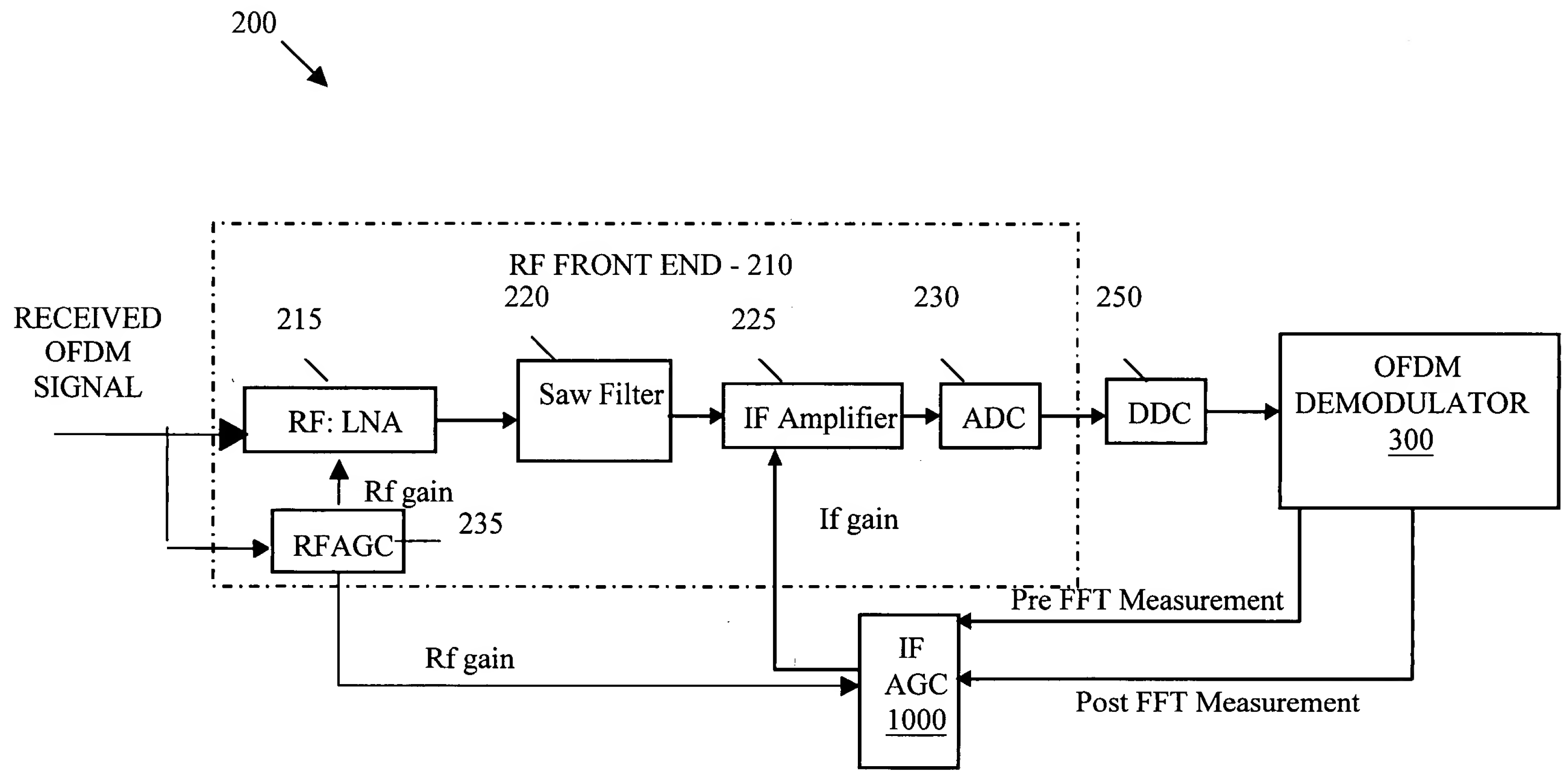

200
RECEIVED OFDM SIGNAL
RF FRONT END - 210
215
RF: LNA
220
Saw Filter
225
IF Amplifier
230
ADC
250
DDC
OFDM DEMODULATOR 300
Rf gain
RFAGC
235
If gain
Pre FFT Measurement
Post FFT Measurement
Rf gain
IF AGC 1000

FIG. 2

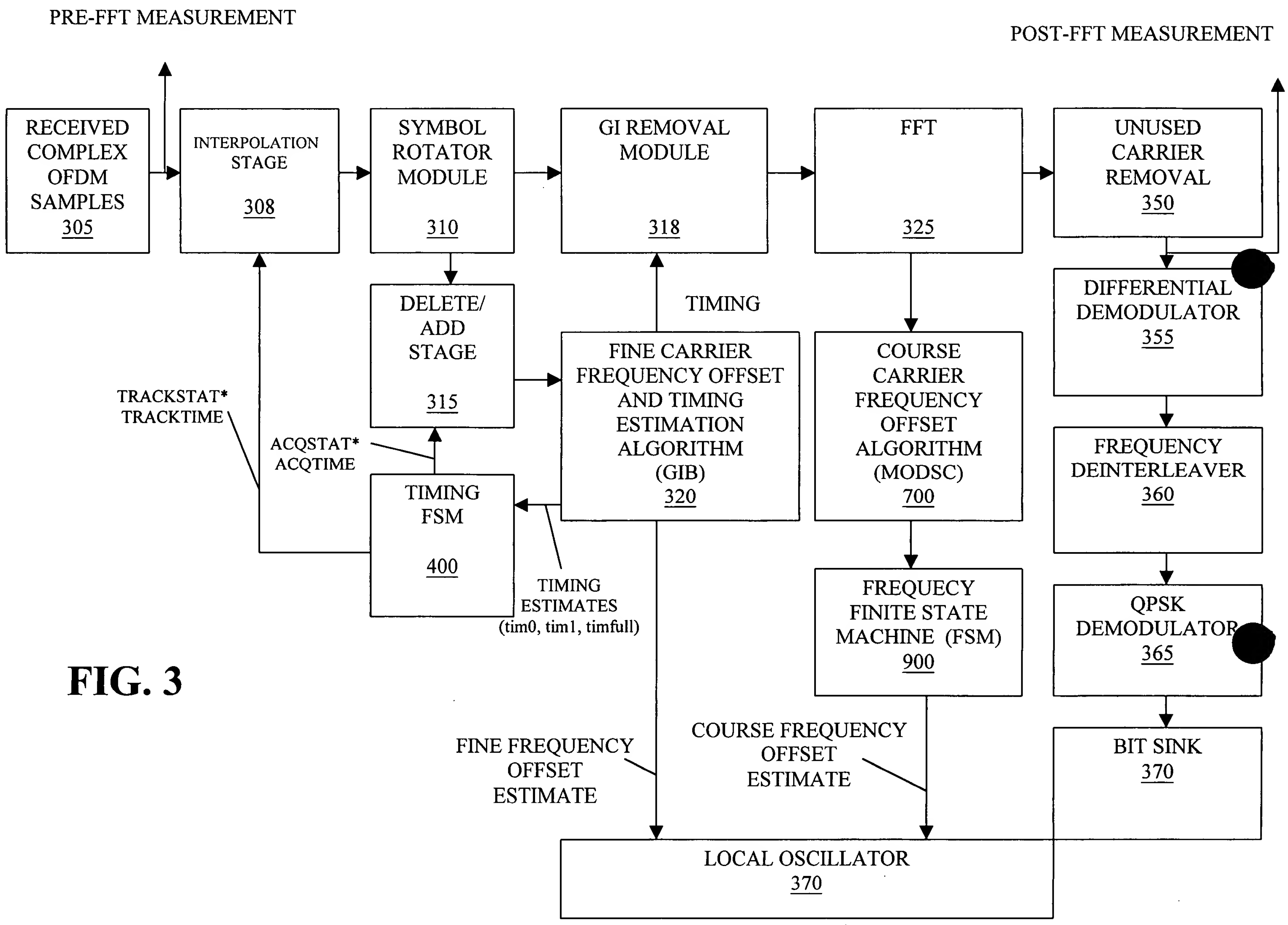

PRE-FFT MEASUREMENT
POST-FFT MEASUREMENT
RECEIVED COMPLEX OFDM SAMPLES 305
INTERPOLATION STAGE 308
SYMBOL ROTATOR MODULE 310
GI REMOVAL MODULE 318
FFT 325
UNUSED CARRIER REMOVAL 350
DIFFERENTIAL DEMODULATOR 355
FREQUENCY DEINTERLEAVER 360
QPSK DEMODULATOR 365
BIT SINK 370
DELETE/ ADD STAGE 315
TIMING
FINE CARRIER FREQUENCY OFFSET AND TIMING ESTIMATION ALGORITHM (GIB) 320
COURSE CARRIER FREQUENCY OFFSET ALGORITHM (MODSC) 700
FREQUECY FINITE STATE MACHINE (FSM) 900
TRACKSTAT* TRACKTIME
ACQSTAT* ACQTIME
TIMING FSM 400
TIMING ESTIMATES (tim0, tim1, timfull)
FINE FREQUENCY OFFSET ESTIMATE
COURSE FREQUENCY OFFSET ESTIMATE
LOCAL OSCILLATOR 370

FIG. 3

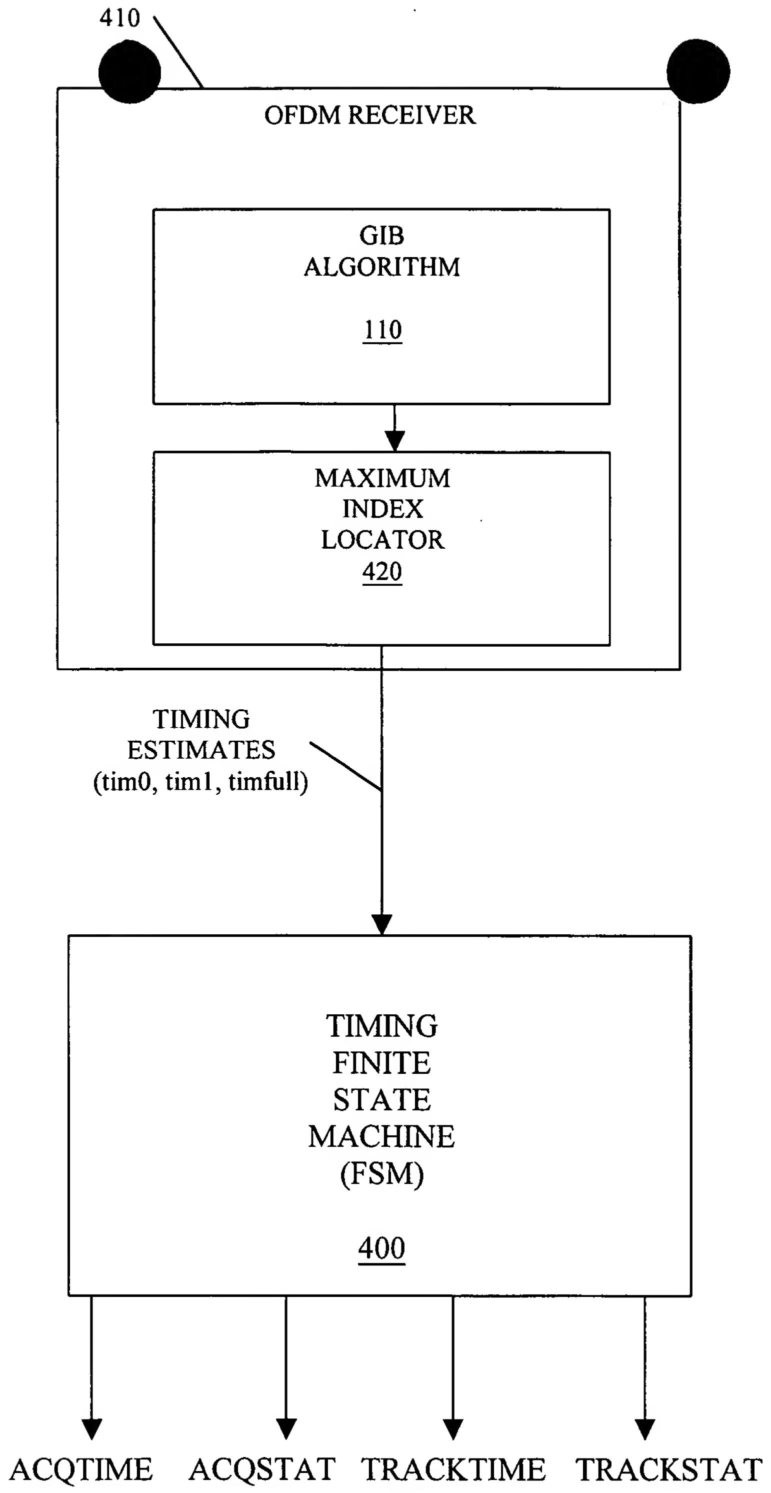

410
OFDM RECEIVER
GIB ALGORITHM
110
MAXIMUM INDEX LOCATOR
420
TIMING ESTIMATES (tim0, tim1, timfull)
TIMING FINITE STATE MACHINE (FSM)
400
ACQTIME
ACQSTAT
TRACKTIME
TRACKSTAT

FIG. 4

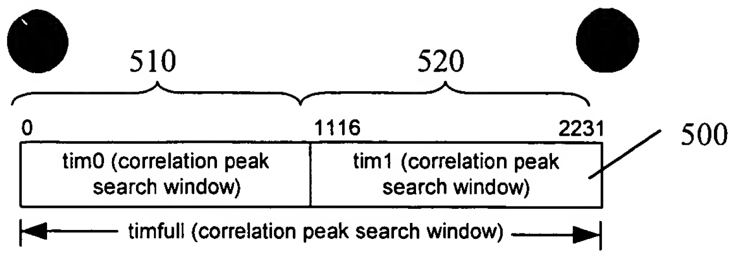

510
520
500
0
1116
2231
tim0 (correlation peak search window)
tim1 (correlation peak search window)
timfull (correlation peak search window)

FIG. 5A

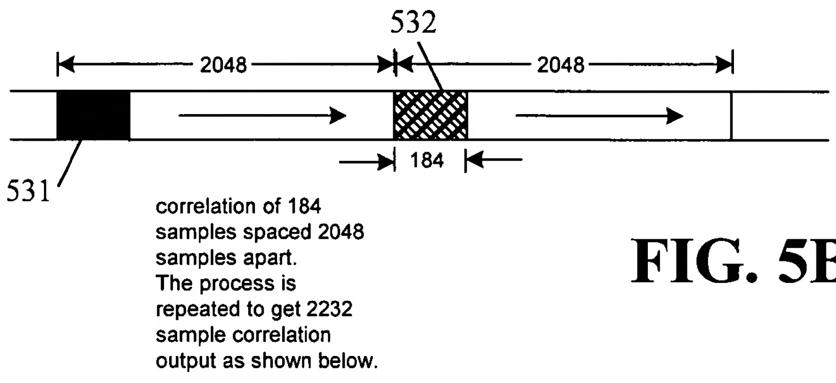

532
2048
2048
184
531
correlation of 184 samples spaced 2048 samples apart. The process is repeated to get 2232 sample correlation output as shown below.

FIG. 5B

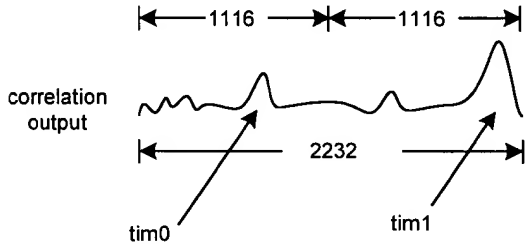


FIG. 5C

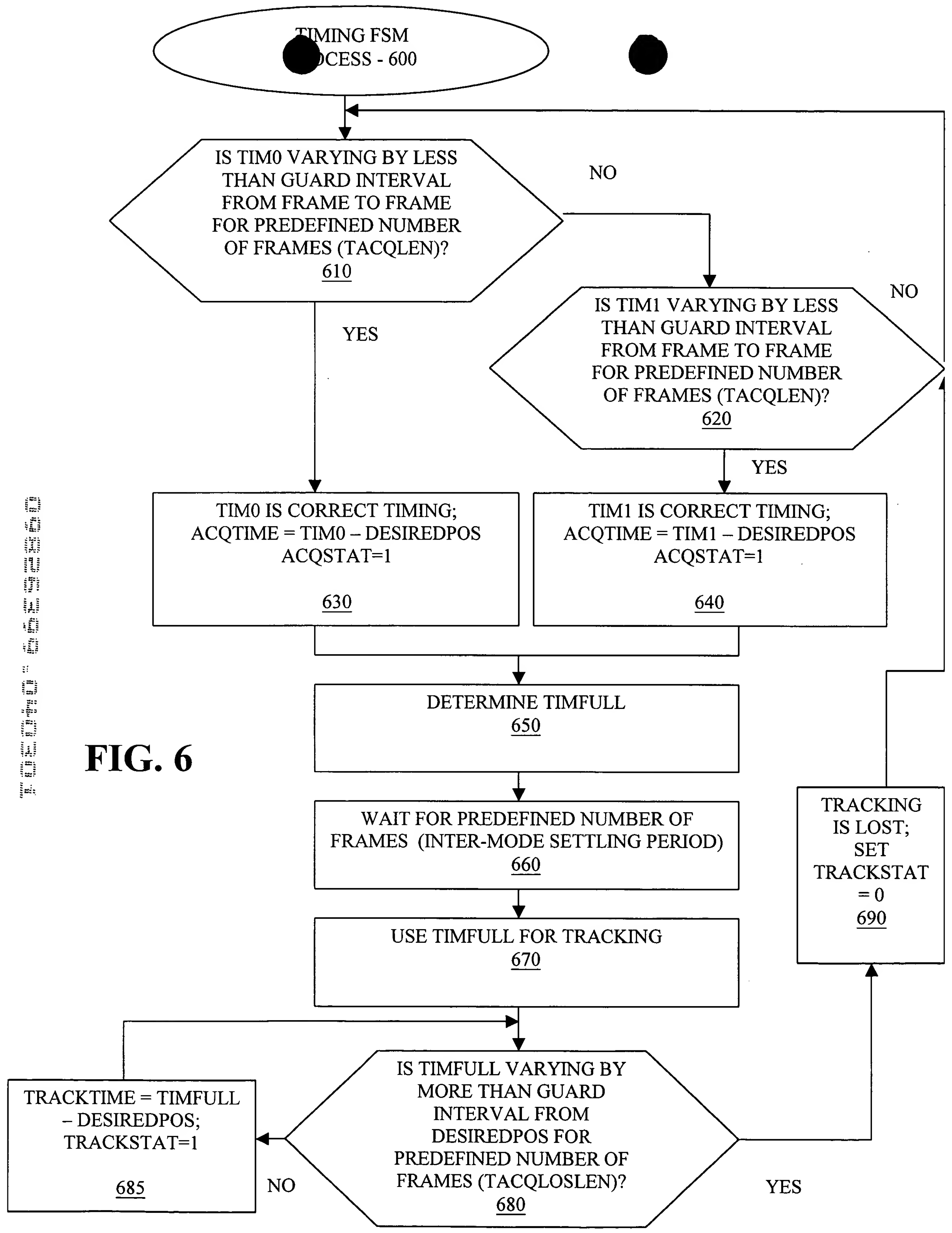

TIMING FSM PROCESS - 600
IS TIM0 VARYING BY LESS THAN GUARD INTERVAL FROM FRAME TO FRAME FOR PREDEFINED NUMBER OF FRAMES (TACQLEN)? 610
NO
YES
IS TIM1 VARYING BY LESS THAN GUARD INTERVAL FROM FRAME TO FRAME FOR PREDEFINED NUMBER OF FRAMES (TACQLEN)? 620
NO
YES
TIM0 IS CORRECT TIMING; ACQTIME = TIM0 – DESIREDPOS ACQSTAT=1 630
TIM1 IS CORRECT TIMING; ACQTIME = TIM1 – DESIREDPOS ACQSTAT=1 640
DETERMINE TIMFULL 650
WAIT FOR PREDEFINED NUMBER OF FRAMES (INTER-MODE SETTLING PERIOD) 660
USE TIMFULL FOR TRACKING 670
IS TIMFULL VARYING BY MORE THAN GUARD INTERVAL FROM DESIREDPOS FOR PREDEFINED NUMBER OF FRAMES (TACQLOSLEN)? 680
NO
YES
TRACKTIME = TIMFULL – DESIREDPOS; TRACKSTAT=1 685
TRACKING IS LOST; SET TRACKSTAT = 0 690

FIG. 6

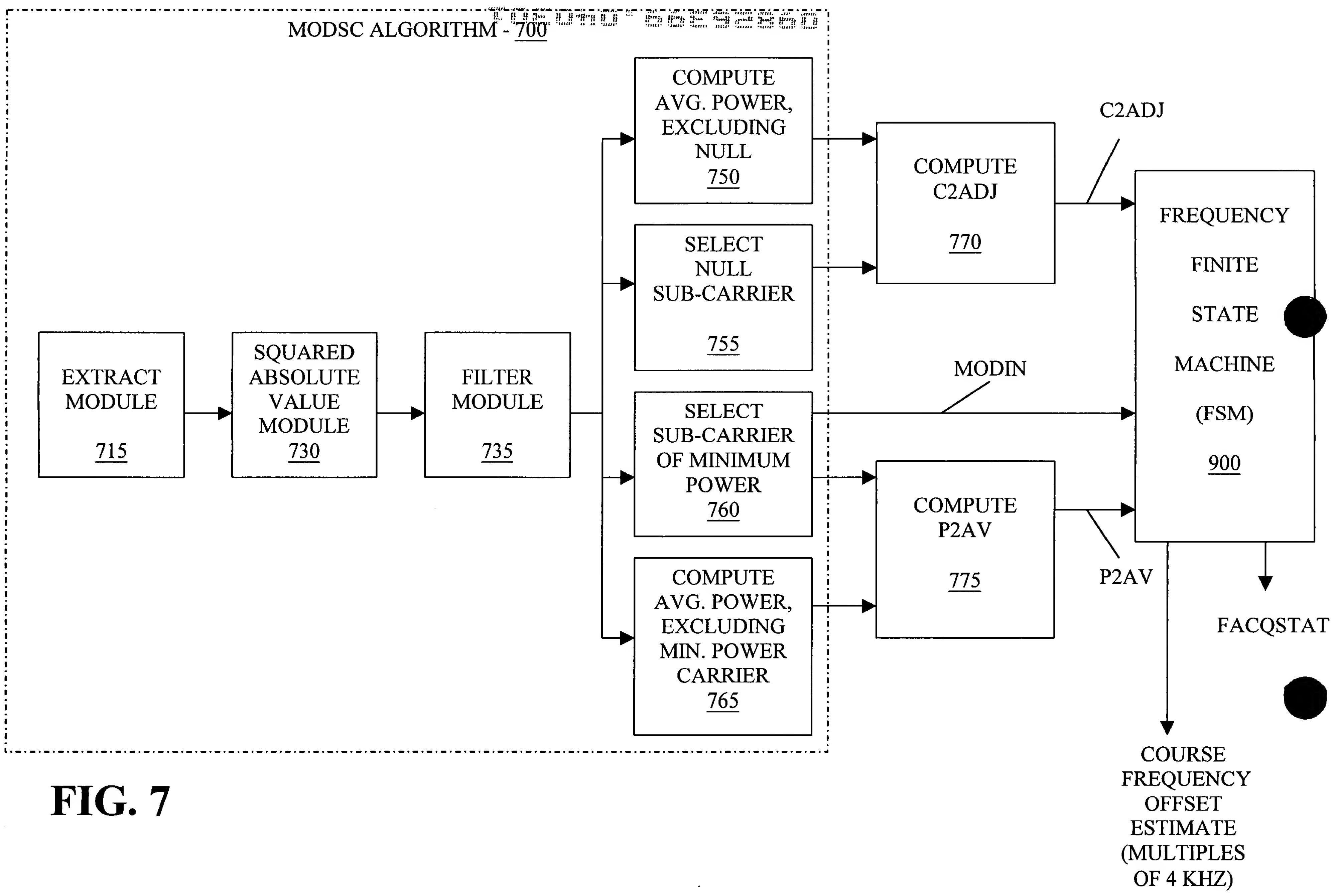

MODSC ALGORITHM - 700
EXTRACT MODULE 715
SQUARED ABSOLUTE VALUE MODULE 730
FILTER MODULE 735
COMPUTE AVG. POWER, EXCLUDING NULL 750
SELECT NULL SUB-CARRIER 755
SELECT SUB-CARRIER OF MINIMUM POWER 760
COMPUTE AVG. POWER, EXCLUDING MIN. POWER CARRIER 765
COMPUTE C2ADJ 770
COMPUTE P2AV 775
C2ADJ
MODIN
P2AV
FREQUENCY FINITE STATE MACHINE (FSM) 900
FACQSTAT
COURSE FREQUENCY OFFSET ESTIMATE (MULTIPLES OF 4 KHZ)

FIG. 7

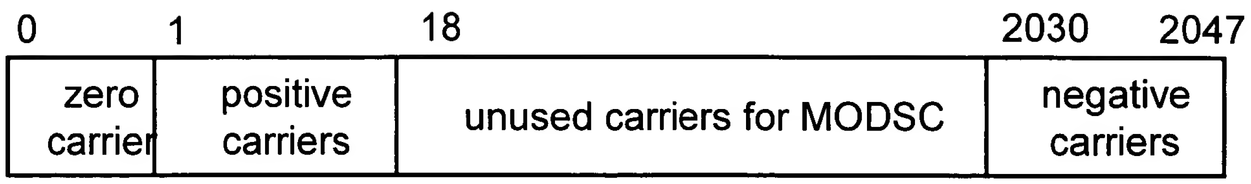
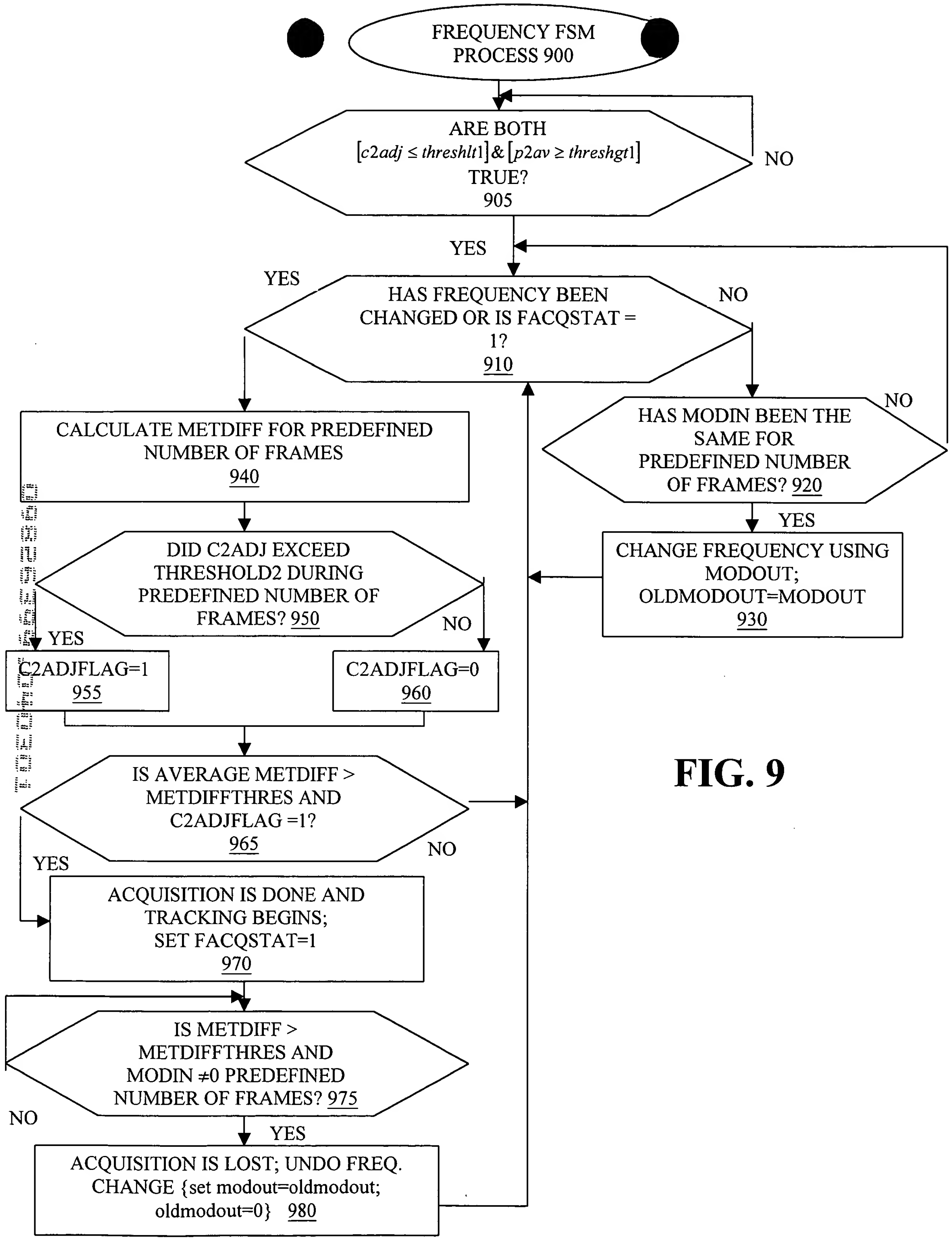

0
1
18
2030
2047
zero carrier
positive carriers
unused carriers for MODSC
negative carriers

FIG. 8



FREQUENCY FSM PROCESS 900
ARE BOTH $[c2adj \leq threshlt1] \& [p2av \geq threshgt1]$ TRUE? 905
NO
YES
HAS FREQUENCY BEEN CHANGED OR IS FACQSTAT = 1? 910
YES
NO
HAS MODIN BEEN THE SAME FOR PREDEFINED NUMBER OF FRAMES? 920
NO
YES
CHANGE FREQUENCY USING MODOUT; OLDMODOUT=MODOUT 930
CALCULATE METDIFF FOR PREDEFINED NUMBER OF FRAMES 940
DID C2ADJ EXCEED THRESHOLD2 DURING PREDEFINED NUMBER OF FRAMES? 950
YES
NO
C2ADJFLAG=1 955
C2ADJFLAG=0 960
IS AVERAGE METDIFF > METDIFFTHRES AND C2ADJFLAG =1? 965
YES
NO
ACQUISITION IS DONE AND TRACKING BEGINS; SET FACQSTAT=1 970
IS METDIFF > METDIFFTHRES AND MODIN ≠0 PREDEFINED NUMBER OF FRAMES? 975
NO
YES
ACQUISITION IS LOST; UNDO FREQ. CHANGE {set modout=oldmodout; oldmodout=0} 980

FIG. 9

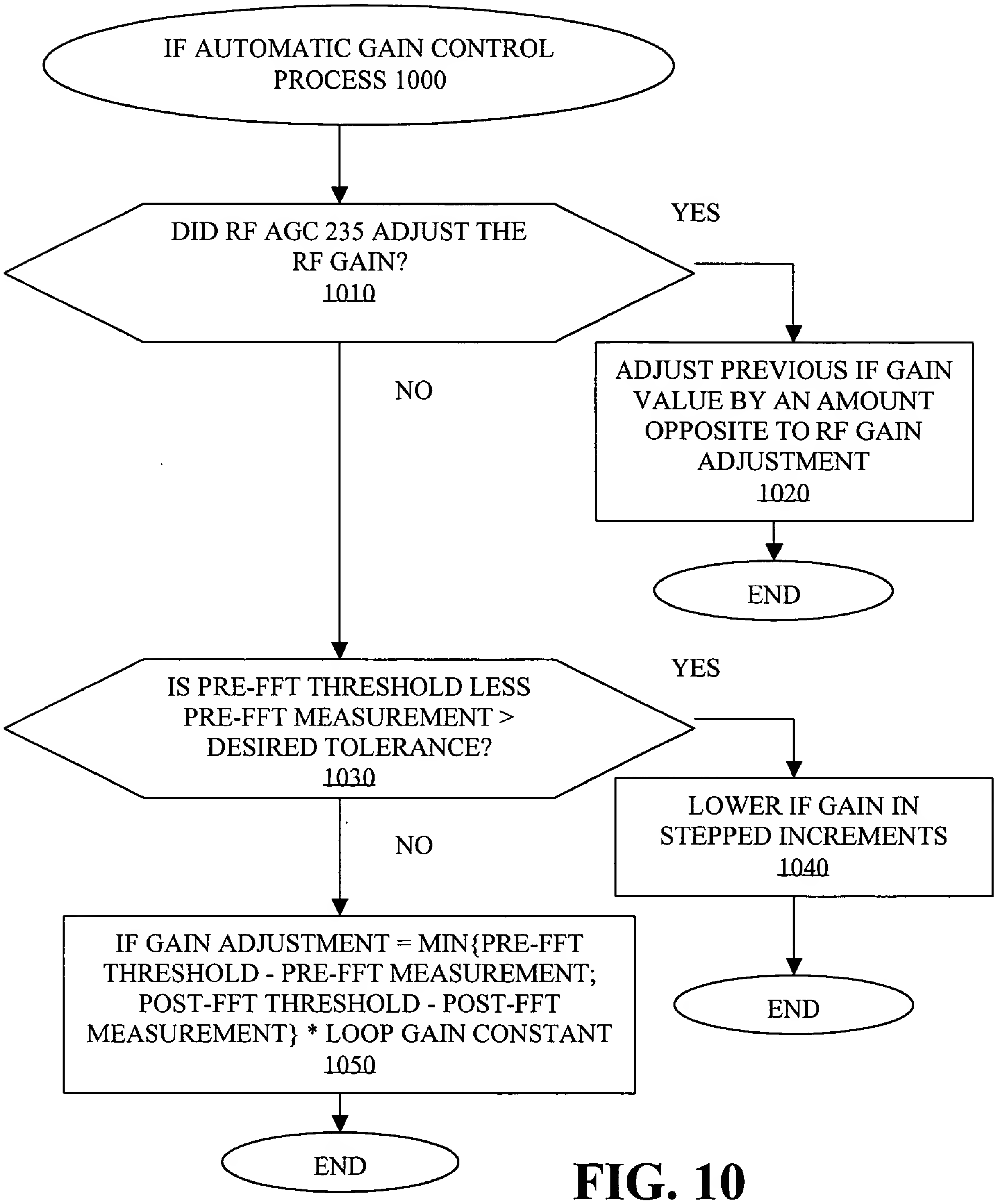

IF AUTOMATIC GAIN CONTROL PROCESS 1000
DID RF AGC 235 ADJUST THE RF GAIN? 1010
YES
ADJUST PREVIOUS IF GAIN VALUE BY AN AMOUNT OPPOSITE TO RF GAIN ADJUSTMENT 1020
END
NO
IS PRE-FFT THRESHOLD LESS PRE-FFT MEASUREMENT > DESIRED TOLERANCE? 1030
YES
LOWER IF GAIN IN STEPPED INCREMENTS 1040
END
NO
IF GAIN ADJUSTMENT = MIN{PRE-FFT THRESHOLD - PRE-FFT MEASUREMENT; POST-FFT THRESHOLD - POST-FFT MEASUREMENT} * LOOP GAIN CONSTANT 1050
END

FIG. 10

```
INPUT_PORT(1)  register float *Prepower;
INPUT_PORT(2)  register float *Postpower;
INPUT_PORT(3)  register float *RFgain;
OUTPUT_PORT(1) register float *Output;  /*IF AGC Gain in dB*/

BLOCKFACTOR long BlockFactor;

PARAMETER(1) float OutputIntervalWidth; /* 71 dB*/
PARAMETER(2) float SetPointdBPre;       /*42.2*/
PARAMETER(3) float SetPointdBPost;      /*32.2*/
PARAMETER(4) float Kagc;          /*0.25*/
PARAMETER(5) float PreDropdB;           /* 3.0*/
PARAMETER(6) long WaitTime;             /*8 OFDM Frames!!*/

STATE float oldoutput;
STATE float oldrfgain;
STATE long counter;

#include <math.h>

void init ofdmagccontrol2()
{
 /* initialize Sum */
 oldoutput = 0.0;
 counter = WaitTime;
}

void ofdmagccontrol2()
{
 register float dbinpre, dbinpost, err, rfgain, output;
 float HalfInterval = (OutputIntervalWidth / 2.0);
```

FIG. 11A

```
LOOP(BlockFactor)

        printf("------------------IFbeg--------\n");
        dbinpre = *Prepower++; dbinpost = *Postpower++;
        rfgain = *RFgain++;

        printf("prepower = %f, post = %f, rfgain = %f\n", dbinpre, dbinpost, rfgain);
        if((rfgain-oldrfgain)!=0.0)
                {
                output = oldoutput -(rfgain-oldrfgain);
                printf("ifgain = -rfdiff = %f, oldrfgain = %f\n", output, oldrfgain);
                }
        else if ((SetPointdBPre-PreDropdB-dbinpre <=0.0)&& (counter >= WaitTime))
                {
                output = oldoutput -(PreDropdB+2.0);
                printf("ifgain = due to Pre  = %f\n", -PreDropdB);
                counter=0;
                }
        else
                {
                counter++;
                if(SetPointdBPre-dbinpre < SetPointdBPost-dbinpost)
                        err = SetPointdBPre - dbinpre;
                else
                        err = SetPointdBPost-dbinpost;
                err = Kagc*err;
                output = oldoutput+err;
                printf("output = %f\n", output);
                }

        if(output>=HalfInterval)
                output  = HalfInterval;
        else if (output<=-HalfInterval)
                output = -HalfInterval;
        else
                output = output;

        *Output++ = output;
        oldrfgain = rfgain;
        oldoutput = output;
        printf("------------------IFend--------\n");
ENDLOOP
}
```

FIG. 11B